

ONTOLOGY-BASED REASONING FOR ENTITY-RELATIONSHIP DATA MODEL SEMANTIC EVALUATION

Zoltan Kazi, Biljana Radulović, Ivana Berković, Ljubica Kazi

Original scientific paper

Conceptual modeling is one of the most important activities in the modeling phase of information systems development most commonly presented by entity-relationship data model. This paper presents a system for entity-relationship data model semantic evaluation that is based on comparing ontology with data model elements. This approach is based on domain ontology and data model formalization at predicate calculus form that is suitable for reasoning. A set of reasoning rules for ontology to data model mapping was defined. The whole process is empirically verified and confirmed. For this purpose it has been developed a software tool for ontology and data model transformation to predicate logic form and then to a set of Prolog-like clauses. After integration of these sets of clauses and rules, a Prolog-system was used for reasoning in order to quantitatively express the quality of data model with appropriate metric.

Keywords: data model; ontology; reasoning rules; semantic evaluation

Ontološki zasnovano rasuđivanje za semantičku evaluaciju modela entiteta-poveznika

Izvorni znanstveni članak

Konceptualno modeliranje je jedna od najbitnijih aktivnosti u fazi modeliranja informacionog sustava najčešće prezentirana modelom entiteta-poveznika. Ovaj rad predstavlja sustav za semantičko evaluiranje modela entiteta-poveznika koji se bazira na komparaciji elemenata ontologije i modela podataka. Ovaj pristup je baziran na formalizaciji domenske ontologije i modela podataka u formu predikatskog računa koji je pogodan za rasuđivanje. Definiran je skup pravila rasuđivanja za mapiranje ontologije u model podataka. Ceo postupak je empirijski verificiran i potvrđen. Za ovu svrhu je razvijen softverski alat koji ontologiju i model podataka transformira u predikatski oblik i zatim u skup Prologolikih klauzula. Nakon integriranja ovih klauzula i pravila uporabljen je Prolog sustav za rasuđivanje sa ciljem da se kvantitativno izrazi kvalitet modela podataka sa odgovarajućom metrikom.

Ključne riječi: model podataka; ontologija; pravila za rasuđivanje; semantička evaluacija

1 Introduction

Building new or improving existing information systems is always a complex project, with many decisions [12] in the design and set of models at different levels of abstraction to manage. Most commonly used models in information system development include analysis phases models (business process models, conceptual data models) and design phase models (data models –entity-relationship, conceptual and relational model, class diagrams, functionality models - use case, sequence, activity diagrams, implementation models - components and deployment diagrams) [9]. Results from analysis are mapped to elements of design. It is very important to focus on quality in results of early phase's development, since cost of removing the same defect that could be allocated in early phases increases significantly if allocated in later phases of development [5].

One of the most important phases in information system development is conceptual data modeling (CDM), since it is a basis for other development activities and project management activities such as software size estimation [4]. Conceptual data models are most commonly presented by entity-relationship (ER) modeling [7]. This is the most widely accepted way of data requirement description at the conceptual level [15].

In the field of information systems models, Van Belle [25] introduces a general metrics framework related to syntax, semantic and pragmatic aspect of model quality evaluation. Data quality research [2] is related to development of methodologies, frameworks and tools for measurement and improvement of data models and data in databases. Results in this field propose frameworks that define the set of quality characteristics, the metrics that

could measure the level of quality characteristics achievement in particular case and the set of activities to perform measurement and metrics for model evaluation.

In recent years, researches in the field of data model evaluation resulted with various proposed methods, metrics and frameworks that are published but a small number is empirically tested [16, 18, 19, 20, 21]. None of these tested methods and metrics are oriented to the semantic validation of a model quality what was the main reason for starting this project and research.

2 Data models evaluation

Methodologies and frameworks for data model quality evaluation could be generally classified as [2]: data-driven vs. process driven methodologies; measurement vs. improvement methodologies and general vs. specific (related to particular model types or notations) methodologies. Batra and Antony [3] presents conceptual modeling errors as human errors at three performance levels: skill-based, rule based and knowledge based.

Research [18] shows analysis of proposed solutions to evaluation of conceptual data models. More than 50 various proposals to conceptual data modeling evaluation are published, but less than 20 % of them are empirically validated. None of proposed solutions is accepted in practice, outside the research environment. These solutions are at different level of generality (the research ones are more general and difficult to be implemented in practice, while the practically motivated ones are more focused on particular modeling notation). The proposed solutions show lack of agreement of terminology, lack of consistency with related fields and standards, lack of measurements metrics and evaluation procedures, lack of

guidelines for improvement (proposed solutions are mostly focused on error detection), lack of attention to process quality (i.e. process of creation of conceptual data models and prevention of errors), but are rather oriented to product quality detection (and some of them: correction), lack of empirical studies from practice (i.e. studies on how conceptual data model evaluation is made in practice). Other empirical validation included action research with collaboration of researchers and practitioners in the field and with practical projects and issues in conceptual data modeling evaluation.

Metrics in [14] are defined with the aim to enable comparison of equivalent models so as to direct designer toward a better design.

Metrics for evaluation of conceptual data models could be classified as:

- Quantitative-based: checking the number of entities, relationships and attributes with certain characteristics [22], complexity of elements and a model [26, 22],
- Qualitative-based: subjective judgment on quality characteristics such as: completeness, integrity, flexibility, comprehensiveness, correctness, simplicity, integration, implement ability [20] and preciseness, completeness, consistency, reliability, timeliness, uniqueness, validity [23],
- Ontology-based [16]: structure-based (suitability, stability, consistency) and content-based (completeness, cohesy, validity),
- Behavioral-based [16]: applicability from user and designer aspect, maintainability, correctness and performances.

Recent researches in the field of automating conceptual data models evaluation consider conceptual data model as a "product". Certain software tools are developed as prototypes that enable:

- Analysis of conceptual data model elements quality, based on domain ontology [24],
- Comparison of created conceptual data model with other models [19],
- Automated reasoning on quality of conceptual data models [8].

Combining action research with practitioners and laboratory research with both experts and novices in conceptual data modeling, progress is made toward generality and applicability of proposed conceptual data model evaluation framework in practice [19]. Still, empirical verification of the proposed framework is subjective in quality criteria metrics ranking, i.e. ranking of created conceptual data models is performed by qualified persons and it is not automated. Recent research results are related to automation in evaluation of conceptual data model [15, 16, 18].

Other prototypes consider process of conceptual data model creation and improve it by enabling assistance or complete automation in:

- Consulting support to novice designers related to conceptual data model elements quality [3],
- Automated creation of conceptual data model design [6].

3 The reasoning system for data model semantic evaluation based on ontology

Motivated by previously presented problems we started a project related to ER data model semantic evaluation. The main idea was integration of automated reasoning system, ontology, data model and reasoning rules with the aim to evaluate the ER data model semantic quality. The ontology is proved to be the adequate technique for dealing with semantic of data [17]. The approach is formulated in the context of data model quality measurement and formal theories mentioned in [13, 14, 16, 18, 26].

3.1 System features and architecture

Our research goal was to develop and empirically verify an automated system for reasoning that will have features such as:

- Rule-based system,
- Enable automated reasoning on ER data model quality.
- Provide answers related to particular element of a created conceptual data model and an overall data model quality evaluation.
- Enable evaluation of semantic aspect of the created ER model and therefore should be based on comparison with "semantically rich" models (such as ontology models [25]) that enable presenting semantic variations.
- Scalable, i.e. should be applicable to any size of the conceptual model.

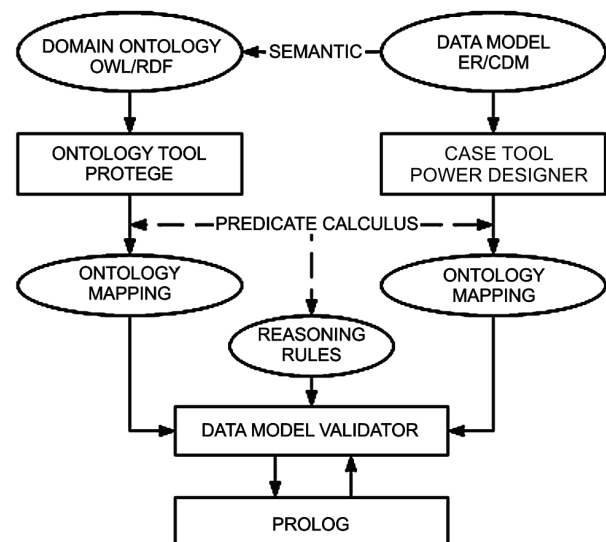


Figure 1 Proposed system for ER data model evaluation

The developed reasoning system consists of several modules, i.e. software tools integrated to a complex system. These modules are:

- Ontology editor/tool for creating ontology,
- CASE tool for creating ER data model,
- Data Model Valuator (DMV) tool for transformation and integration of ontology and ER data model into formal language sentences,
- Prolog as a core reasoning system that computes answers to queries.

3.2 Ontology and data model formalization

Data model is a formal abstraction through which the real world is mapped in the database [25]. It enables representation of a real world concepts and elements through a set of data entities and their connections. They can be represented in various ways: graphical representation with schemas, data dictionary representation and formal languages representation, such as predicate logic calculus.

Formal presentation of ER data model is extension of formalization presented in [11] where data model is represented as $S = (E, A, R, C, P)$, where:

- E is a finite set of entities,
- A is a finite set of attributes,
- R is a finite set of relationships,
- C is a finite set of constraints concerning domain, definition, relationships and semantics associated to the elements and attributes,
- P is a finite set of association rules among entities, attributes, relationships and constraints.

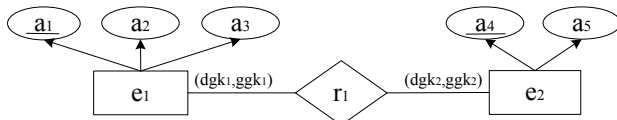


Figure 2 ER data model schema

Formalization of an ER model includes creating sets of elements that are written as Prolog-like clauses. Predicate names for elements of S set are: *ent* for E set, *atr* for A set, *rel* for R set, *res* for C set and *p* for P set. Set of formalized elements of conceptual data model schema from Fig. 2 is listed below:

$$E = \{ent(e_1), ent(e_2)\}$$

$$A = \{atr(a_1), atr(a_2), atr(a_3), atr(a_4), atr(a_5)\}$$

$$R = \{rel(r_1)\}$$

$$S = \{res(idatr), res(mandatory), res(dgk_1), res(dgk_2), res(ggk_1), res(ggk_2), res(dom_1), res(dom_2), res(dom_3), res(dom_4), res(dom_5)\}$$

$$P = \{p(e_1, a_1), p(e_1, a_2), p(e_1, a_3), p(e_2, a_4), p(e_2, a_5), p(e_1, idatr), p(e_2, idatr), p(a_1, mandatory), p(a_2, mandatory), p(a_4, mandatory), p(a_5, mandatory), p(e_1, r_1), p(r_1, e_2), p(dgk_1, r_1), p(r_1, dgk_2), p(ggk_1, r_1), p(r_1, ggk_2), p(a_1, dom_1), p(a_1, idatr), p(a_2, dom_2), p(a_4, idatr), p(a_3, dom_3), p(a_4, dom_4), p(a_5, dom_5)\}$$

where:

- *idatr* is identifying attribute,
- $dom_1, dom_2, dom_3, dom_4, dom_5$ are attributes data types (domains),
- *mandatory* is a sign for mandatory attributes,
- dgk_1, dgk_2 , are lower cardinality restrictions,
- ggk_1, ggk_2 are upper cardinality restrictions.

Ontology is often used to capture and share knowledge in a specific domain of interest [24]. Ontology describes the concepts in the domain and also the relationships that hold between those concepts [25]. The basic characteristics of ontology are hierarchy of concepts/objects, which is established by using different semantic links [13]. Ontology elements like type, class, subclass, property, sub-property, domain and range could be mapped to predicate logic form according to [1]. Predicate logic form of ontology could be written in Prolog-like form like ER model elements. Structure of ontology is a collection of OWL/RDF elements that are transformed into RDF expression as a collection of triplets, each consisting of subject, predicate and objects [27]. Facts that are described with RDF triplets represent a relation between things denoted by subject and object of the triplet, or even their properties: RDF (Subject, Predicate, Object).

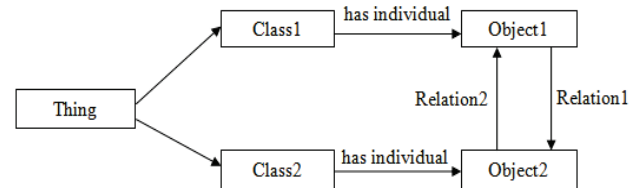


Figure 3 Domain ontology scheme

Mapping RDF/OWL ontology elements into Prolog-like clauses considered an *rdf* name for predicate name in Prolog system.

The element *class1* from abstract ontology is shown in Fig. 3 that has an individual named *object1*. This class and instance is presented with two triplets:

```
S = subj:'Class1',
P = rdf:type,
O = owl:'Class';
R(S,P,O) → rdf(class1, type, class).
```

```
S = subj:Object1,
P = rdf:type,
O = subj:'NamedIndividual';
R(S,P,O) → rdf(object1, type, namedindividual).
```

List of these predicates for an abstract ontology shown in Fig. 3 is listed below:

```
rdf(class1, type, class).
rdf(class2, type, class).
rdf(relation1, type, object property).
rdf(relation2, type, object property).
rdf(object1, type, named individual).
rdf(object2, type, named individual).
rdf(object1, relation1, object2).
rdf(object2, relation2, object1).
```

3.3 Reasoning rules

Model evaluation in this system is performed by applying a set of reasoning rules to formalized representation of ER data model and ontology with the aim to compare them.

Mapping ontology to data model elements is based on research [10], where:

- Ontology class is mapped to entity type,
- Ontology data property is mapped to attribute,
- Ontology data property range is mapped to attribute data type/domain,
- Ontology object property is mapped to relationship,
- Ontology property constraint is mapped to relationship property (cardinality, dependency).

The reasoning rules for evaluation of ontology-to-conceptual data model mapping are:

A) Mapping classes from ontology to entities in ER data model:

Rule 1 - Ontology classes that are covered by entities in ER model. For each class from ontology must be defined named entity set in data model [10]:

$$ontoclassent(x) : \neg rdf(x, type, class), ent(x). \quad (1)$$

Rule 2 - Ontology classes that are not covered by entities in ER data model:

$$ontoclassentno(x) : \neg rdf(x, type, class), notent(x). \quad (2)$$

B) Mapping ontology data properties to attributes in ER data model:

Rule 3 - Data properties from ontology that are covered by attributes in ER data model. For each data property in ontology must be defined named attribute in data model [10]:

$$ontodataattrib(x) : \neg rdf(x, type, dataproperty), atr(x). \quad (3)$$

Rule 4 - Ontology data properties that are not covered by attributes in ER data model:

$$ontodataattribno(x) : \neg rdf(x, type, dataproperty), notatr(x). \quad (4)$$

Rule 5 - Data properties and data properties ranges from ontology that are covered by attributes with defined data types in conceptual data model. For each attribute in data model from set of attributes there is a restriction with data type name [10]:

$$ontodataattribtype(x, y) : \neg rdf(x, type, dataproperty), rdf(x, range, y), atr(x), res(z), p(x, z), datatype(y, z). \quad (5)$$

C) Mapping ontology object properties to relationships in ER data model:

Rule 6 - Object properties from ontology that are covered by relationships in conceptual data model. For each object property from ontology must be declared named relationship in ER data model:

$$ontoobjproprel(x) : \neg rdf(x, type, objectproperty), rel(x). \quad (6)$$

Rule 7 - Ontology object properties that are not covered by relationships in conceptual data model:

$$ontoobjpropnorel(x) : \neg rdf(x, type, objectproperty), notrel(x). \quad (7)$$

Rule 8 - Ontology object properties that are covered by relationships in conceptual data model that are defined between entities that match appropriate ontology classes:

$$\begin{aligned} ontorel(xc_1, yop, xc_2, xe_1, yr, xe_2) : & - \\ & rdf(xc_1, type, class), rdf(xo_1, classassertion, xc_1), \\ & rdf(xo_1, type, namedindividual), \\ & rdf(xc_2, type, class), rdf(xo_2, classassertion, xc_2), \\ & rdf(xo_2, type, namedindividual), \\ & rdf(yop, type, objectproperty), rdf(xo_1, yop, xo_2), \\ & ent(xe_1), ent(xe_2), rel(yr), p(xe_1, yr), p(yr, xe_2), \\ & p(y, x_2), xe_1 = xc_1, xe_2 = xc_2, yop = yr. \end{aligned} \quad (8)$$

Rule 9 - Ontology object property ranges that are covered by relationship cardinality in conceptual data model that are defined between entities that match appropriate ontology classes:

$$\begin{aligned} ontorel(xc, yop, zcd_1, zcd_2) : & - \\ & rdf(yop, type, objectproperty), rdf(xc, yop, zcd_1), \\ & rdf(xc, yop, zcd_2), ent(xe_1), ent(xe_2), rel(yr), \\ & p(xe_1, yr), p(yr, xe_2), \left(\begin{aligned} & (p(zcd_1, yr), p(zcd_2, yr)), \\ & (p(yr, zcd_1), p(yr, zcd_2)) \end{aligned} \right), \\ & (xe_1 = xc, xe_2 = xc), yop = yr, not zcd_1 = zcd_2. \end{aligned} \quad (9)$$

D) Mapping classes and subclasses from ontology to IS_A hierarchy relationship in ER data model:

Rule 10 - Ontology classes and subclasses that are covered by IS_A hierarchy entities in conceptual data model. According to [10] for each class from ontology must be defined a named entity super-class type in data model, and each ontology subclass is presented with entity subtype, with restriction that subtypes in data model must be different objects:

$$\begin{aligned} ontosubclassisa(x, x_1, x_2) : & \neg rdf(x, type, class), \\ & rdf(x_1, subclass, x), rdf(x_2, subclass, x), ent(x), \\ & ent(x_1), ent(x_2), p(x, y), p(y, x_1), p(y, x_2), \\ & not x_1 = x_2. \end{aligned} \quad (10)$$

Rule 11 - Ontology classes and subclasses that are not covered by IS_A hierarchy entities in conceptual data model. For each ontology class must be defined named entity super-class type in data model, and each ontology subclass is not presented with entity subtype:

$$\begin{aligned} & \text{ontosubclassnoisa}(x, x_1, x_2) : \neg \text{rdf}(x, \text{type}, \text{class}), \\ & \text{rdf}(x_1, \text{subclass}, x), \text{rdf}(x_2, \text{subclass}, x), \text{not } x_1 = x_2, \quad (11) \\ & \text{notontosubclassnoisa}(x, x_1, x_2) \end{aligned}$$

Explanation of symbols used in reasoning rules:

- $x, x_1, x_2, xc, xc_1, xc_2, xe_1, xe_2, y, yop, yr, z, zcd_1$, and zcd_2 represents variables,
- $\text{type}, \text{class}, \text{subclass}, \text{objectproperty}, \text{dataproperty}, \text{range}$, and classassertion represents constant values,
- $\text{rdf}, \text{ent}, \text{atr}, \text{rel}, \text{res}$, and p represents predicate symbols.

3.4 Ontology mark calculation for a data model

For each ER data model final rank evaluation from the aspect of ontology mapping (OM) is quantitatively represented as a sum of ontology mapping evaluation points for each element of the data model. These particular marks for elements are measured by handling the Prolog answers on goals. For each data element is given a "weight factor" K_T , where T represents an ER element type. Weight factor, according to [16], represents a quantitative expressed significance of an element in the analysis of the whole conceptual data model.

An ontology point for entities is calculated as:

$$OM_E = \frac{\sum_{i_E=1}^{n_E} E(\text{Rule1})_{i_E} \cdot 100}{\sum_{i_E=1}^{n_E} E(\text{Rule1})_{i_E} + \sum_{i_E=1}^{n_E} E(\text{Rule2})_{i_E}}, \quad (12)$$

An ontology point for attributes is calculated as:

$$OM_A = \frac{\frac{\sum_{i_A=1}^{n_A} A(\text{Rule3})_{i_A} + \sum_{i_A=1}^{n_A} A(\text{Rule5})_{i_A}}{2} \cdot 100}{\sum_{i_A=1}^{n_A} A(\text{Rule3})_{i_A} + \sum_{i_A=1}^{n_A} A(\text{Rule4})_{i_A}}, \quad (13)$$

An ontology point for relationships is calculated as:

$$OM_R = \frac{\frac{\sum_{i_R=1}^{n_R} R(\text{Rule6})_{i_R} + \sum_{i_R=1}^{n_R} R(\text{Rule8})_{i_R} + \sum_{i_R=1}^{n_R} R(\text{Rule9})_{i_R}}{3} \cdot 100}{\sum_{i_R=1}^{n_R} R(\text{Rule6})_{i_R} + \sum_{i_R=1}^{n_R} R(\text{Rule7})_{i_R}}, \quad (14)$$

An ontology point for classes and subclasses is calculated as:

$$OM_{SC} = \frac{\sum_{i_{SC}=1}^{n_{SC}} SC(\text{Rule10})_{i_{SC}} \cdot 100}{\sum_{i_{SC}=1}^{n_{SC}} SC(\text{Rule10})_{i_{SC}} + \sum_{i_{SC}=1}^{n_{SC}} SC(\text{Rule11})_{i_{SC}}}, \quad (15)$$

Total ontology mark for entire ER data model is calculated as:

$$OM = \frac{K_E \cdot OM_E + K_A \cdot OM_A + K_R \cdot OM_R + K_{SC} \cdot OM_{SC}}{4}, \quad (16)$$

Explanation for Eqs. (12) ÷ (16) elements:

- OM is ontology points for each data model,
- OM_E is ontology points for entities,
- OM_A is ontology points for attributes,
- OM_R is ontology points for relationships,
- OM_{SC} is ontology points for super-classes entities and sub-classes entities,
- K_E, K_A, K_R, K_{SC} are weight factors.

Minimum values for OM, OM_E, OM_A, OM_R and OM_{SC} particular marks are 0, while maximum value could be 100 for particular and also for total ontology mark for a whole data model.

3.5 Process of using the system

The proposed system is implemented by using the following software tools:

- Ontology editor Protégé developed at Stanford University for creating ontology.
- CASE tool Sybase Power Designer for projecting ER/conceptual data model.
- Amzi! Prolog as a reasoning system that computes answers to queries.

For the purpose of files transformation and integration to appropriate Prolog program needed for Amzi! Prolog, special Data Model Valuator (DMV) tool was created by using Microsoft Visual Studio.NET development environment. The process of using this tool starts with creating ontology by using an ontology editor. The ER model is created in a CASE tool. DMV tool could be started. A user could start an option for loading ER model and an option for formalization of data model that will parse elements of data model to a set of Prolog-like clauses and present them in user interface. Another option is loading ontology for its transformation to a set of Prolog-like clauses that are also presented. Third step is loading a set of defined reasoning rules. An example of using DMV tool up to this point is presented in Fig. 4 with created ontology (Fig. 6) and ER data model (Fig. 5) for this system usability empirical validation.

After all clauses are created and ready in integrated list (i.e. conceptual model's clauses, ontology's clauses and reasoning rules), we used Prolog as a core reasoning system for computation of answers to queries related to particular data model and ontology. Answers from reasoning system must be included in previously defined metrics (12), (13), (14), (15) and (16) for ER data model semantic evaluation.

Example of query with answers for ontology from empirical study with nine classes that has to be covered by appropriate entity types in ER data model (Rule 1):

?- ontoclassent(X).

X = conference,
 X = fee,
 X = city,
 X = paper,
 X = room,
 no
 Number of goals: 5

X = national,
 X = organizer,
 X = participant,
 no
 Number of goals: 4

An ontology mark for entities is calculated by Eq. (10):

$$OM_E = (5 \cdot 100) / (5 + 4) = 500 / 9 = 55,55.$$

On this mode must be calculated ontology marks for all elements of ER model by (12), (13), (14), (15) and then the final ontology mark for entire ER model by (16).

Query in Prolog and answers for listing all ontology classes that are not covered by entity types in ER data model (Rule 2):

?- ontoclassentno(X).

X = international,

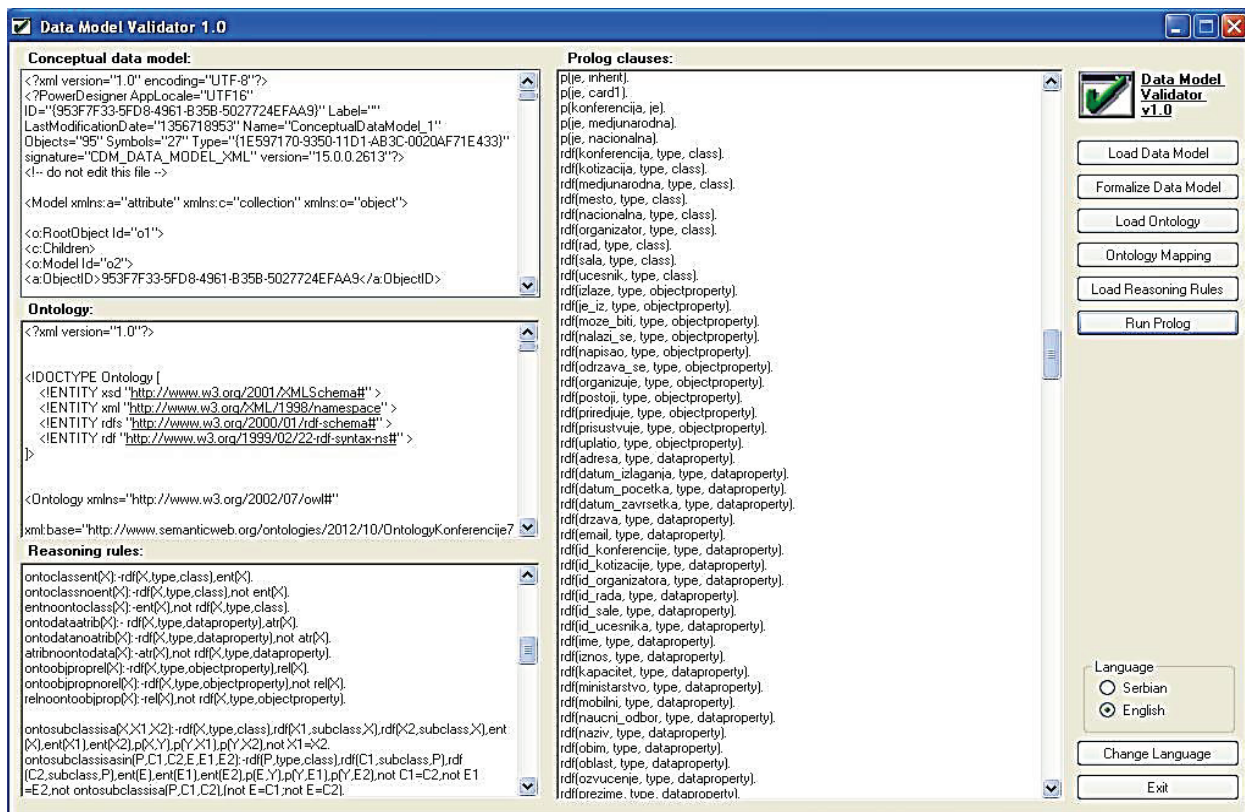


Figure 4 Data Model Validator software tool

4 Empirical study

4.1 Research methodology

The initial empirical testing of the system has been made with a case study in application of initial set of reasoning rules to a single ER data model. Empirical research is conducted as a laboratory experiment with students' data models collected from the practical exam. Participants in this research are students from University of Novi Sad, Technical faculty "Mihajlo Pupin" in Zrenjanin, Serbia. They are all students of the second year of undergraduate (bachelor) studies of information technology engineering.

These 132 participants were given the same exam, i.e. a textual specification of a case study for organizing international conferences (shown in Fig. 5). A single ontology is created to represent the specified case study and domain of problem area (shown in Fig. 6).

Each of students' data models was loaded in DMV tool to be integrated with ontology and set of reasoning rules presented from Eq. (1) to Eq. (11). Integrated programs were individually loaded in the Amzi! Prolog listener environment for executing queries according to rules (1) to (11). Prolog listener has shown results of each query answer computation as we presented for rules (1) and (2). After mapping ontology in empirical study with DMV tool into the Prolog-like clauses we create over 330 facts in RDF triplets. Students' data models result with minimally 160 to more than 250 facts in Prolog sentences. Integrated program for reasoning with rules has from 500 to almost 600 clauses that were all individually loaded into the AMZI! Prolog to be processed.

Statistics is performed upon all results data used for overall evaluation of each ER data model by using equation (14) and $K_T = 1$ (which means that each "weight factor" is 1 for any of evaluated model, i.e. all considered are equally significant).

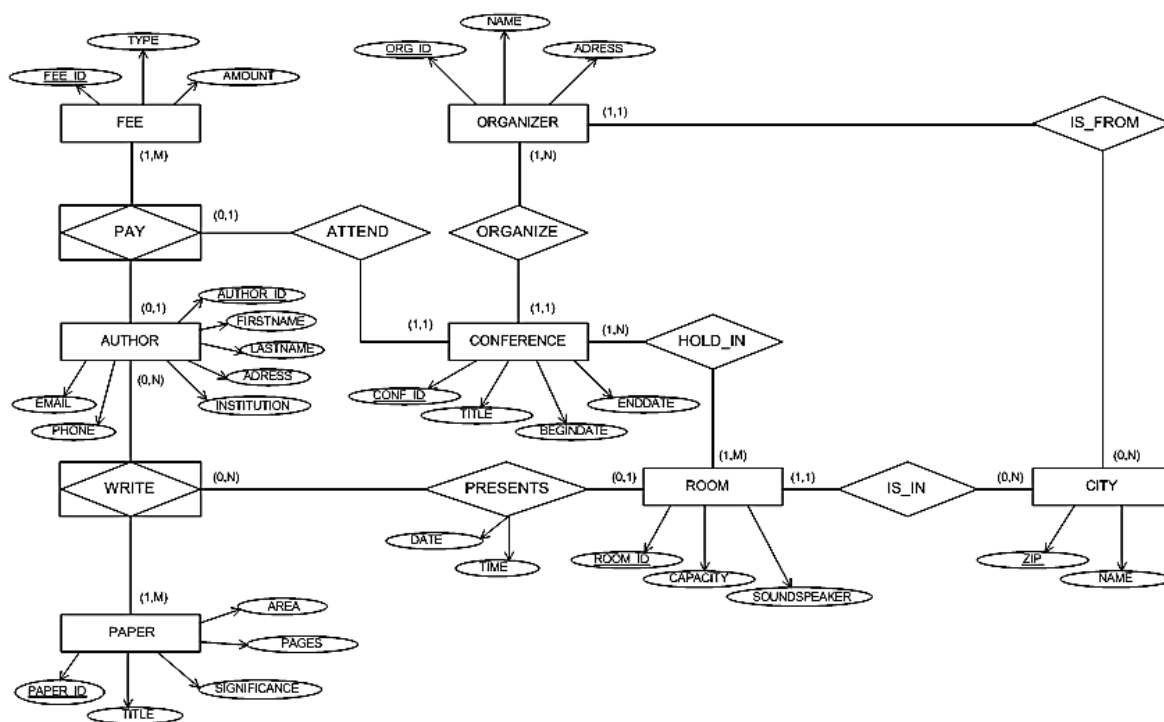


Figure 5 ER data model schema that was used in empirical study

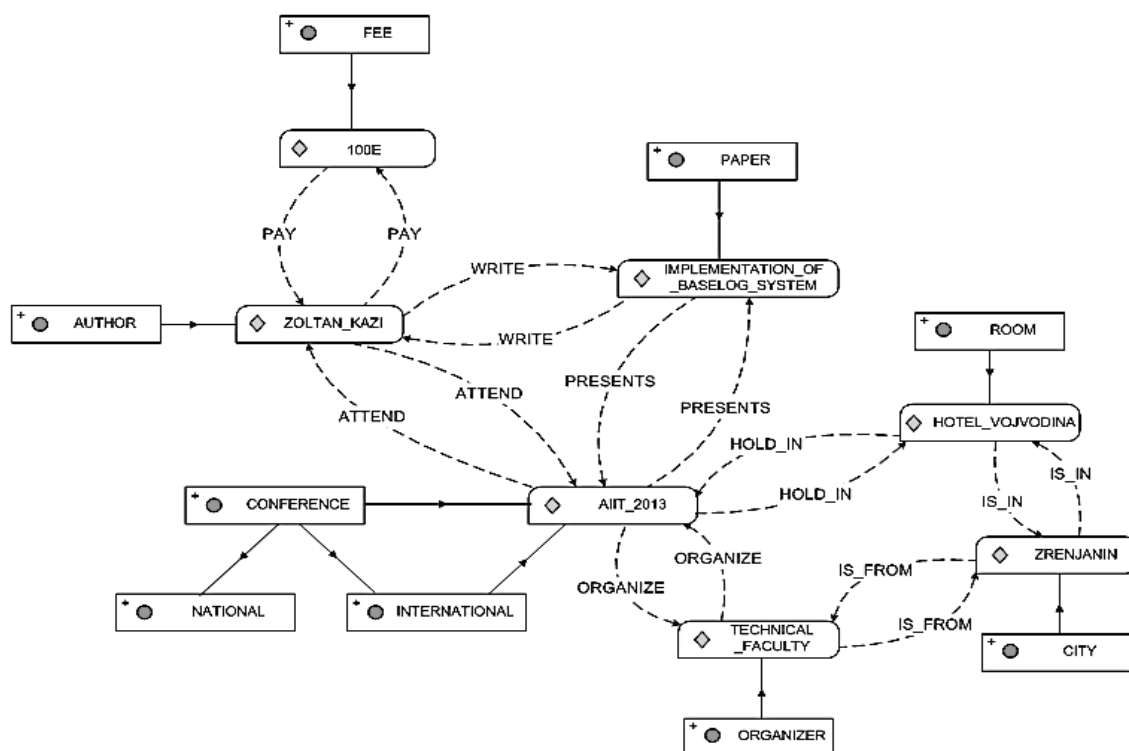


Figure 6 Domain ontology used in empirical study

4.2 Empirical results

Overall statistics related to each reasoning rule accomplishment in all models is presented in Tab.1. Analysis of statistics on empirical results shows that ontology classes are covered by entities in ER data model with more than 92 %, ontology data properties are covered with 54 % appropriate attributes, while object properties are covered by relationships in ER model with 41 %. Ontology classes are covered by only 30 % of appropriate super-class type entities. At the end it can be

seen that ontology sub-classes are covered by 30 % subtype entities. Ontology data properties and data property ranges are covered by 41 % of attributes and data types in data model. A result of computation of each model's ontology mapping evaluation mark is presented in Tab. 2. From the sample of 132 models that have been tested five data models are shown that have the greatest semantic completeness and suitability to ontology and the five worst created data models.

Analysis of empirical results for each ER data model ontology mapping evaluation shows that the best models

do not have better than 89 % of evaluation points, while the worst done models are approximately at 35 %. Average result of all tested and evaluated data models is

almost 64 % of semantically correctness, i.e. completeness and suitability to domain ontology.

Table 1 Empirical results for data model elements semantic evaluation

Elements from ontology and data model	Average number of elements per model	Total number of elements in ontology	Ontological elements coverage (%)
Ontology classes covered by entities in data model	8,34	9	92,68
Ontology data properties covered by attributes in data model	16,3	30	54,33
Ontology data properties and data property ranges covered by attributes and data types in data model	12,43	30	41,43
Ontology object properties covered by relationships in data model	3,1	9	34,44
Ontology classes covered by super-class type entities in data model	0,3	1	30,00
Ontology subclasses covered by subtype entities in data model	0,6	2	30,00

Table 2 Empirical results for ER data model semantic evaluation (the best five and the worst five models score)

Ontological mark for entities	Ontological mark for attributes	Ontological mark for relationships	Ontological mark for classes and subclasses	Total ontological mark for data model
100	68	89	100	89,25
100	68	78	100	86,50
100	64	78	100	85,50
100	59	78	100	84,25
100	57	78	100	83,75
78	44	44	0	41,50
78	43	44	0	41,25
78	51	33	0	40,50
67	42	44	0	38,25
78	32	33	0	35,50

5 Conclusion

From introduction of ER modeling as conceptual data modeling methodology, many research efforts have been focused on creating methodologies and frameworks for evaluation of conceptual data model evaluation, especially in the last decade. Still there is no consensus in creating a unique or integrated framework or standard in this field. Most of the proposed frameworks are still in the domain of theory and less than 20 % of them are empirically evaluated.

This paper shows results in a project of developing a reasoning system for ER data model evaluation based on domain ontology. This system integrates results of using CASE tool for data model creation, ontology editor for ontology creation, reasoning rules for data model evaluation based on mapping with ontology within an automated reasoning system that computes answers needed for metric. Overview of the system is presented, with theoretical contribution that is reflected in formalization of data model and mapping with ontology in a form of clauses. Results of empirical testing and verification of developed system are given.

There are many contributions of the presented research. It has been shown that it is possible to evaluate a semantic aspect of an ER data model. The proposed solution is based on mapping of data model with ontology. This approach is applicable in situations where ontology is created as a basis for evaluation of a group of data models related to the same semantics. This system is scalable and flexible, with ability to separate reasoning rules from reasoning logic. Within DMV tool, an extended formal representation of ER data model is implemented.

Future work could include adapting system to other types of data models, extension of reasoning rules to enable both syntax and semantic verification, with the aim to enable more complete data model verification. This system must be empirically tested with large data models. One further step could be development of consultation expert module that would provide presentation of conceptual data modeling errors and suggestions to improvements.

6 References

- [1] Antoniou, G.; Damasio, C. V.; Grosz, B.; Horrocks, I.; Kifer, M.; Maluszynski, J.; Patel-Schneider, P. F. Combining Rules and Ontologies - A Survey. EU Project: Reasoning on the Web with Rules and Semantics. 2005, (11.05.2011).
- [2] Batini, C.; Scannapieco, M. Data Quality. Springer, 2006.
- [3] Batra, D.; Antony, S. R. Consulting Support During Conceptual Database Design in the Presence of Redundancy in Requirements Specifications: An Empirical Study. // International Journal of Human-Computer Studies, Elsevier. 54(2001), pp. 25-51. DOI: 10.1006/ijhc.2000.0406
- [4] Beng, H.; Tan, K.; Zhao, Y.; Zhang, H. Conceptual Data Model - Based Software Size Estimation for Information Systems. // ACM Transactions on Software Engineering and Methodology. 19, 2(2009), Article 4.
- [5] Boehm, B. W. Software Engineering Economics, Prentice-Hall Inc., Englewood Cliffs, NJ, USA, 1981.
- [6] Brdjanin, D.; Maric, S. An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram. // Computer Science and Information Systems COMSIS. 9, 1(2012), pp. 250-283.
- [7] Chen, P. The Entity-Relationship Data Model – Toward a Unified View of Data. // ACM Transactions on Database Systems. 1, 1(1976), pp. 9-36. DOI: 10.1145/320434.320440

- [8] Choppella, V.; Sengupta, A.; Robertson, E. L.; Johnson, S. D. Preliminary Explorations in Specifying and Validating Entity-Relationship Models in PVS. // Automated Formal Methods Conference AFM / Atlanta, GA, USA, 2007.
- [9] De Lucia, A.; Gravino, C.; Oliveto, R.; Tortora, G. An Experimental Comparison of ER and UML Class Diagrams for Data Modelling. // Journal of Empirical Software Engineering, Springer Science+Business Media, LLC, 2009.
- [10] El-Ghalayini, H.; Odeh, M.; McClatchey, R.; Solomonides, T. Reverse Engineering Ontology to Conceptual Data Models. // 23rd IASTED International Conference on Databases and Applications, DBA / Innsbruck, Austria, 2005.
- [11] Emer, M. C.; Vergilio, S. R.; Jino, M. Testing Relational Database Schemas with Alternative Instance Analysis. // 20th International Conference on Software Engineering & Knowledge Engineering SEKE / San Francisco, USA, 2008.
- [12] Filip, F. G. A Decision-Making Perspective for Designing and Building Information Systems // International Journal of Computers, Communications & Control. 7, 2(2012), pp. 264-272. DOI: 10.15837/ijccc.2012.2.1408
- [13] Formica, A.; Missikoff, M. Correctness of ISA Hierarchies in Object-Oriented Database Schemas. // Advances in Database Technology / Springer-Verlag Berlin Heidelberg, 2006.
- [14] Gray, R.; Carey, B.; McGlynn, N.; Pengelly, A. Design Metrics for Database Systems. // BT Technology Journal. 9, 4(1991), pp. 69-79.
- [15] Jukić, B.; Jukić, N.; Nestorov, S. Process and Data Logic Integration: Logical Links between UML Use Case Narratives and ER Diagrams // Journal of Computing and Information Technology CIT. 21, 3(2013), pp. 161-170. DOI: 10.2498/cit.1002192
- [16] Kesh, S. Evaluating the Quality of Entity Relationship Models. // Information and Software Technology. 37, 12(1995), pp. 681-689. DOI: 10.1016/0950-5849(96)81745-9
- [17] Lammari, N.; Métais, E. Building and Maintaining Ontologies: a Set of Algorithms. // Data & Knowledge Engineering. 48, 2(2004), pp. 155-176. DOI: 10.1016/S0169-023X(03)00103-4
- [18] Moody, D. Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions // Data & Knowledge Engineering. 55, 3(2005), pp. 243-276. DOI: 10.1016/j.datak.2004.12.005
- [19] Moody, D.; Shanks, G. Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework. // Information Systems. 28, 6(2003), pp. 619-650. DOI: 10.1016/S0306-4379(02)00043-1
- [20] Moody, D. Metrics For Evaluating the Quality of Entity Relationship Models. // Proceedings of the 17th International Conference on Conceptual Modelling / Singapore, (1998), pp. 213-225. DOI: 10.1007/978-3-540-49524-6_18
- [21] Piattini, M.; Genero, M.; Calero, C. Data Model Metrics. // UK Academy of Information Systems Annual Conference / Cardiff, UK, 2000.
- [22] Piattini, M.; Genero, M.; Calero, C.; Polo, M.; Ruiz, F. Database Quality. // In Chapter 14: Advanced Database Technology and Design, Eds. Mario Piattini and Oscar Díaz / Artech House, 2000, pp. 485-509.
- [23] Piprani, B.; Ernst, D. A Model for Data Quality Assessment. // Proceedings of OTM Workshops / Springer-Verlag Berlin Heidelberg, 2008, pp. 750-759. DOI: 10.1007/978-3-540-88875-8_99
- [24] Sugumaran, V.; Storey, V. C. The Role of Domain Ontologies in Database Design: An Ontology Management and Conceptual Design Environment // ACM Transactions on Database Systems. 31, 3(2006), pp. 1064-1094. DOI: 10.1145/1166074.1166083
- [25] Ullman, J.; Garcia-Molina, H.; Widom, J. Database Systems: The Complete Book. Department of Computer Science, Stanford University, Prentice Hall, New Jersey, USA, 2002.
- [26] Van Belle, J. P. A Framework for the Evaluation of Business Models and its Empirical Validation. // The Electronic Journal Information Systems Evaluation. 9, 1(2006), pp. 31-44. (20.05.2011).
- [27] W3C Resource Description Framework (RDF): Concepts and Abstract Syntax. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210> (29.05.2011).

Authors' addresses

Zoltan Kazi, PhD, assistant professor

University of Novi Sad
Technical faculty "Mihajlo Pupin" Zrenjanin
Djure Djakovica bb, 23000 Zrenjanin, Serbia
E-mail: zoltan.kazi@gmail.com

Biljana Radulović, PhD, full professor

University of Novi Sad
Technical faculty "Mihajlo Pupin" Zrenjanin
Djure Djakovica bb, 23000 Zrenjanin, Serbia
E-mail: biljana.radulovic@tfzr.rs

Ivana Berković, PhD, full professor

University of Novi Sad
Technical faculty "Mihajlo Pupin" Zrenjanin
Djure Djakovica bb, 23000 Zrenjanin, Serbia
E-mail: berki@sbb.rs

Ljubica Kazi, PhD, teaching assistant

University of Novi Sad
Technical faculty "Mihajlo Pupin" Zrenjanin
Djure Djakovica bb, 23000 Zrenjanin, Serbia
E-mail: ljubicakazi@hotmail.com